

# A stochastic coordinate descent splitting primal-dual fixed point algorithm and applications to large-scale composite optimization

Meng Wen<sup>1</sup>, Yu-Chao Tang<sup>2</sup>, Jigen Peng<sup>1</sup>

1. School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049,  
P.R. China

2. Department of Mathematics, NanChang University, Nanchang 330031, P.R. China

**Abstract** In this paper, we consider the problem of finding the minimizations of the sum of two convex functions and the composition of another convex function with a continuous linear operator from the view of fixed point algorithms based on proximity operators, which is inspired by recent results of Chen, Huang and Zhang. With the idea of coordinate descent, we design a stochastic coordinate descent splitting primal-dual fixed point algorithm. Based on randomized krasnosel'skii mann iterations and the firmly nonexpansive properties of the proximity operator, we achieve the convergence of the proposed algorithms. Moreover, we give two applications of our method. (1) In the case of stochastic minibatch optimization, the algorithm can be applied to split a composite objective function into blocks, each of these blocks being processed sequentially by the computer. (2) In the case of distributed optimization, we consider a set of  $N$  networked agents endowed with private cost functions and seeking to find a consensus on the minimizer of the aggregate cost. In that case, we obtain a distributed iterative algorithm where isolated components of the network are activated in an uncoordinated fashion and passing in an asynchronous manner. Finally, we illustrate the efficiency of the method in the framework of large scale machine learning applications. Generally speaking, our method SCDSPDFP<sup>2</sup>O is comparable with other state-of-the-art methods in numerical performance, while it has some advantages on parameter selection in real applications.

---

\* Corresponding author.

E-mail address: wen5495688@163.com

**Keywords:** fixed point algorithm; coordinate descent; proximity operator; distributed optimization

**MR(2000) Subject Classification** 47H09, 90C25,

## 1 Introduction

In this paper, we aim at solving the following minimization problem

$$\min_{x \in \mathcal{X}} f(x) + g(x) + (h \circ D)(x), \quad (1.1)$$

where  $\mathcal{X}$  and  $\mathcal{Y}$  are two Euclidean spaces,  $f, g \in \Gamma_0(\mathcal{X})$ ,  $h \in \Gamma_0(\mathcal{Y})$ , and  $f$  is differentiable on  $\mathcal{Y}$  with a  $1/\beta$ -Lipschitz continuous gradient for some  $\beta \in (0, +\infty)$  and  $D : \mathcal{X} \rightarrow \mathcal{Y}$  a linear transform. This parameter  $\beta$  is related to the convergence conditions of algorithms presented in the following section. Here and in what follows, for a real Hilbert space  $\mathcal{H}$ ,  $\Gamma_0(\mathcal{H})$  denotes the collection of all proper lower semi-continuous convex functions from  $\mathcal{H}$  to  $(-\infty, +\infty]$ . Despite its simplicity, when  $g = 0$  many problems in image processing can be formulated in the form of (1.1). For instance, the following variational sparse recovery models are often considered in image restoration and medical image reconstruction:

$$\min \frac{1}{2} \|Ax - b\|_2^2 + \lambda \psi(Dx), \quad (1.2)$$

where  $\|\cdot\|_2$  denotes the usual Euclidean norm for a vector,  $A \in \mathbb{R}^{p \times n}$  describes a blur operator,  $b \in \mathbb{R}^p$  represents the blurred and noisy image and  $\lambda > 0$  is the regularization parameter in the context of deblurring and denoising of images.

For problem (1.2), Chen et al proposed a primal-dual fixed point algorithm (*PDFP<sup>2</sup>O*) in [1], i.e.

$$\begin{cases} v_{n+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} f_1})(D(x_n - \gamma \nabla f_2(x_n)) + (I - \lambda D D^T)v_n), \\ x_{n+1} = x_n - \gamma \nabla f_2(x_n) - \lambda D^T v_{n+1}, \end{cases} \quad (1.3)$$

where  $0 < \lambda \leq 1/\lambda_{\max}(D D^T)$ ,  $0 < \gamma < 2\beta$ , and the operator  $\text{prox}_f$  is called the proximity operator of  $f$ . Note that this type of splitting method was originally studied in [10,11] and the notion of proximity operators was first introduced by Moreau in

[12] as a generalization of projection operators. Motivated and inspired by the above results, we introduced a splitting primal-dual fixed point algorithm. The contributions of us are the following aspects:

(I) The algorithm that we proposed includes the well known PFPS [13] and  $FP^2O$  [14] as a special case. Moreover, the idea based on the results of Chen et al [1], and the obvious advantage of the proposed scheme is that it is very easy for parallel implementation.

(II) Based on the results of Chen et al [1] and Bianchi et al [2], we introduce the idea of stochastic coordinate descent on splitting primal-dual fixed point algorithm. The form of splitting primal-dual fixed point algorithm can be translated into fixed point iterations of a given operator having a nonexpansive property. By the view of stochastic coordinate descent, we know that at each iteration, the algorithm is only to update a random subset of coordinates. Although this leads to a perturbed version of the initial splitting primal-dual fixed point iterations, but it can be proved to preserve the convergence properties of the initial unperturbed version. Moreover, stochastic coordinate descent has been used in the literature [15-17] for proximal gradient algorithms. We believe that its application to splitting primal-dual fixed point algorithm well suited to large-scale optimization problems.

(III) We use our views to large-scale optimization problems which arises in signal processing and machine learning contexts. We prove that the general idea of stochastic coordinate descent gives a unified framework allowing to derive stochastic algorithms of different kinds. Furthermore, we give two application examples. Firstly, we propose a new stochastic approximation algorithm by applying stochastic coordinate descent on the top of SPDFP<sup>2</sup>O. The algorithm is called as stochastic minibatch splitting primal-dual fixed point algorithm (SMSPDFP<sup>2</sup>O). Secondly, we introduce a random asynchronous distributed optimization methods that we call as distributed asynchronous splitting primal-dual fixed point algorithm (DASMSPDFP<sup>2</sup>O). The algorithm can be used to efficiently solve an optimization problem over a network of communicating agents. The algorithms are asynchronous in the sense that some components of the network are allowed to wake up at random and perform local updates, while the rest of the network stands still. No coordinator or global clock is needed. The frequency of activation of the various network components is likely to vary.

The rest of this paper is organized as follows. In the next section, we introduce some notations used throughout in the paper. In section 3, we devote to introduce SPDFP<sup>2</sup>O algorithm and its relation with the PDFP<sup>2</sup>O, we also show how the SPDFP<sup>2</sup>O includes PDFP<sup>2</sup>O as a special case. In section 4, we propose a stochastic approximation algorithm from the SPDFP<sup>2</sup>O. In section 5, we address the problem of asynchronous distributed optimization. In the final section, we show the numerical performance and efficiency of propose algorithm through some examples in the context of large-scale  $l_1$ -regularized logistic regression.

## 2 Preliminaries

Throughout the paper, we denote by  $\langle \cdot, \cdot \rangle$  the inner product on  $\mathcal{X}$  and by  $\| \cdot \|$  the norm on  $\mathcal{X}$ . We consider the case where  $D$  is injective (in particular, it is implicit that  $\dim(\mathcal{X}) \leq \dim(\mathcal{Y})$ ). In the latter case, we denote by  $\mathcal{R} = \text{Im}(D)$  the image of  $D$  and by  $D^{-1}$  the inverse of  $D$  on  $\mathcal{R} \rightarrow \mathcal{X}$ . We emphasize the fact that the inclusion  $\mathcal{R} \subset \mathcal{Y}$  might be strict. We denote by  $\nabla$  the gradient operator. We make the following assumptions:

**Assumption 2.1.** *The following facts holds true:*

- (1)  $D$  is injective;
- (2)  $f$  has  $1/\beta$ -Lipschitz continuous gradient.

**Assumption 2.2.** *The infimum of problem (1.1) is attained. Moreover, the following qualification condition holds*

$$0 \in \text{ri}(\text{dom } h - D \text{ dom } g).$$

**Definition 2.1.** Let  $f$  be a real-valued convex function on  $\mathcal{X}$ , the operator  $\text{prox}_f$  is defined by

$$\begin{aligned} \text{prox}_f : \mathcal{H} &\rightarrow \mathcal{H} \\ x &\mapsto \arg \min_{y \in H} f(y) + \frac{1}{2} \|x - y\|_2^2, \end{aligned}$$

called the proximity operator of  $f$ .

**Definition 2.2.** Let  $A$  be a closed convex set of  $\mathcal{X}$ . Then the indicator function of  $A$  is defined as

$$\iota_A(x) = \begin{cases} 0, & \text{if } x \in A, \\ \infty, & \text{otherwise.} \end{cases}$$

It can easy see the proximity operator of the indicator function in a closed convex subset  $A$  can be reduced a projection operator onto this closed convex set  $A$ . That is,

$$\text{prox}_{\iota_A} = \text{proj}_A$$

where  $\text{proj}$  is the projection operator of  $A$ .

**Definition 2.3.** (Nonexpansive operators and firmly nonexpansive operators [4]). An operator  $T : \mathcal{H} \rightarrow \mathcal{H}$  is nonexpansive if and only if it satisfies

$$\|Tx - Ty\|_2 \leq \|x - y\|_2 \text{ for all } (x, y) \in \mathcal{H}^2.$$

$T$  is firmly nonexpansive if and only if it satisfies one of the following equivalent conditions:

- (i)  $\|Tx - Ty\|_2^2 \leq \langle Tx - Ty, x - y \rangle$  for all  $(x, y) \in \mathcal{H}^2$ ;
- (ii)  $\|Tx - Ty\|_2^2 = \|x - y\|_2^2 - \|(I - T)x - (I - T)y\|_2^2$  for all  $(x, y) \in \mathcal{H}^2$ .

It is easy to show from the above definitions that a firmly nonexpansive operator  $T$  is nonexpansive.

**Lemma 2.1.** (Lemma 2.4 of [3]). Let  $f$  be a function in  $\Gamma_0(\mathcal{X})$ . Then  $\text{prox}_f$  and  $I - \text{prox}_f$  are both firmly nonexpansive operators.

For an element  $u = (v, x) \in \mathcal{Y} \times \mathcal{X}$ , with  $v \in \mathcal{Y}$  and  $x \in \mathcal{X}$ , let

$$\|u\|_\lambda = \sqrt{\|x\|_2^2 + \lambda\|v\|_2^2}.$$

We can easily see that  $\|\cdot\|_\lambda$  is a norm over the produce space  $\mathcal{Y} \times \mathcal{X}$  whenever  $\lambda > 0$ .

**Lemma 2.2.** ([1]). Let Assumptions 2.2 hold true. If  $0 < \gamma < 2\beta$ ,  $0 < \lambda \leq 1/\lambda_{\max}(\tilde{D}\tilde{D}^T)$ , Let  $(\tilde{v}^{k+1}, x^{k+1}) = T(\tilde{v}^k, x^k)$  where  $T$  is the transformation described by Equations (3.3). Then  $T$  is nonexpansive under the norm  $\|\cdot\|_\lambda$ .

**Definition 2.4.** (Randomized krasnosel'skii mann iterations[2]). Let  $\mathcal{V}$  be a Euclidean space. Consider the space  $\mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_J$  for some  $J \in \mathbb{N}^*$  where for any  $j$ ,  $\mathcal{V}_j$  is a Euclidean space. For  $\mathcal{V}$  equipped with the scalar product  $\langle x, y \rangle = \sum_{j=1}^J \langle x_j, y_j \rangle_{\mathcal{V}_j}$  where  $\langle \cdot, \cdot \rangle_{\mathcal{V}_j}$  is the scalar product in  $\mathcal{V}_j$ . For  $j \in \{1, \dots, J\}$ , let  $T_j : \mathcal{V} \rightarrow \mathcal{V}_j$  be the components of the output of operator  $T : \mathcal{V} \rightarrow \mathcal{V}$  corresponding to  $\mathcal{V}_j$ , so, we have  $Tx = (T_1x, \dots, T_Jx)$ . Let  $2^{\mathcal{J}}$  be the power set of  $\mathcal{J} = \{1, \dots, J\}$ . For any  $\kappa \in 2^{\mathcal{J}}$ , we denote the operator  $\hat{T}^\kappa : \mathcal{V} \rightarrow \mathcal{V}$  by  $\hat{T}_j^\kappa x = T_jx$  for  $j \in \kappa$  and  $\hat{T}_j^\kappa x = x_j$  for otherwise. On some probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ , we introduce a random i.i.d. sequence  $(\zeta^k)_{k \in \mathbb{N}^*}$  such that  $\zeta^k : \Omega \rightarrow 2^{\mathcal{J}}$  i.e.  $\zeta^k(\omega)$  is a subset of  $\mathcal{J}$ . Assume that the following holds:

$$\forall j \in \mathcal{J}, \exists \kappa \in 2^{\mathcal{J}}, j \in \kappa \quad \text{and} \quad \mathbb{P}(\zeta_1 = \kappa) > 0. \quad (2.1)$$

**Lemma 2.3.** (Theorem 3 of [2]). Let  $T : \mathcal{V} \rightarrow \mathcal{V}$  be  $\alpha$ -averaged and  $\text{Fix}(T) \neq \emptyset$ . Let  $(\zeta^k)_{k \in \mathbb{N}^*}$  be a random i.i.d. sequence on  $2^{\mathcal{J}}$  such that Condition (2.1) holds. If for all  $k$ , sequence  $(\beta_k)_{k \in \mathbb{N}}$  satisfies

$$0 < \liminf_k \beta_k \leq \limsup_k \beta_k < \frac{1}{\alpha}.$$

Then, almost surely, the iterated sequence

$$x^{k+1} = x^k + \beta_k (\hat{T}^{(\zeta^{k+1})} x^k - x^k) \quad (2.2)$$

converges to some point in  $\text{Fix}(T)$ .

In particular, if  $T$  is nonexpansive, and for all  $k$ , sequence  $(\beta_k)_{k \in \mathbb{N}}$  satisfies

$$0 < \liminf_k \beta_k \leq \limsup_k \beta_k < 1.$$

We can know the iterated sequence (2.2) converges to some point in  $\text{Fix}(T)$ .

### 3 Splitting primal-dual fixed point algorithm

When  $g = 0$ , for problem (1.1) Chen et al [1] considered a primal-dual fixed point algorithm based on the proximity operator( $PDFP^2O$ ) as follows:

$$\begin{cases} v^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda}h})(D(x^k - \gamma \nabla f(x^k)) + (I - \lambda DD^T)v^k), \\ x^{k+1} = x^k - \gamma \nabla f(x^k) - \lambda D^T v^{k+1}, \end{cases} \quad (3.1)$$

where  $0 < \gamma < 2\beta$ ,  $0 < \lambda \leq 1/\lambda_{\max}(DD^T)$ ,  $\lambda_{\max}(DD^T)$  is the largest eigenvalue of  $DD^T$ ,  $I$  is identity operator or unit matrix.

The convergence of  $PDFP^2O$  is guaranteed by the following theorem.

**Theorem 3.1.** ([1]) Suppose  $0 < \gamma < 2\beta$  and  $0 < \lambda \leq 1/\lambda_{\max}(DD^T)$ . Let  $u_k = (v_k, x_k)$  be the sequence generated by  $PDFP^2O$ . Then the sequence  $\{x_k\}$  converges to a solution of problem (1.1).

Similar to the primal-dual fixed point algorithm based on proximity operator (PDFP<sup>2</sup>O), we proposed an algorithm called SPDFP<sup>2</sup>O to solve (1.1) as follows:

---

**Algorithm 1** Splitting primal-dual fixed points algorithm based on proximity operator (SPDFP<sup>2</sup>O).

---

Initialization: Choose  $x^0, y^0 \in \mathcal{X}$ ,  $v^0 \in \mathcal{Y}$ ,  $0 < \lambda \leq 1/(\lambda_{\max}(DD^T) + 1)$ ,  $0 < \gamma < 2\beta$ .

Iterations ( $k \geq 0$ ): Update  $x^k, v^k, x^{k+\frac{1}{2}}$  as follows

$$\begin{cases} x^{k+\frac{1}{2}} = x^k - \gamma \nabla f(x^k), \\ v^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda}h})(Dx^{k+\frac{1}{2}} + (I - \lambda DD^T)v^k - \lambda Dy^k), \\ y^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda}g})(x_{k+\frac{1}{2}} + (I - \lambda)y^k - \lambda D^T v^k), \\ x^{k+1} = x_{k+\frac{1}{2}} - \lambda D^T v^{k+1} - \lambda y^{k+1}. \end{cases}$$

end for

---

**Theorem 3.2.** Suppose  $0 < \gamma < 2\beta$  and  $0 < \lambda \leq 1/(\lambda_{\max}(DD^T) + 1)$ . Let  $u_k = (v_k, x_k)$  be the sequence generated by SPDFP<sup>2</sup>O. Then the sequence  $\{x_k\}$  converges to a solution of problem (1.1).

*Proof.* By setting  $\tilde{D} = (D, I)^T$ ,  $\tilde{h}(v, y) = h(v) + g(y), \forall (v, y) \in \mathcal{Y} \times \mathcal{X}$ , we have  $(\tilde{h} \circ \tilde{D})(x) = h(Dx) + g(x), \forall x \in \mathcal{X}$ . So, the problem (1.1) can be formulated as follows:

$$\min_{x \in \mathcal{X}} f(x) + (\tilde{h} \circ \tilde{D})(x), \quad (3.2)$$

Based on the reference[1], we can obtain the following iterative sequence:

$$\begin{cases} \tilde{v}^{k+1} = (I - \text{prox}_{\tilde{\lambda}\tilde{h}})(\tilde{D}(x^k - \gamma\nabla f(x^k)) + (I - \lambda\tilde{D}\tilde{D}^T)\tilde{v}^k), \\ x^{k+1} = x^k - \gamma\nabla f(x^k) - \lambda\tilde{D}^T\tilde{v}^{k+1}, \end{cases} \quad (3.3)$$

where  $0 < \gamma < 2\beta$ ,  $0 < \lambda \leq 1/(\lambda_{\max}\tilde{D}\tilde{D}^T) = 1/(\lambda_{\max}(DD^T) + 1)$ ,  $\tilde{v}_k = (v_k, y_k)^T$ . Since the function  $\tilde{h}$  is separable with the variables  $v, y$ , then the formula (3.3) is equivalent to

$$\begin{cases} v^{k+1} = (I - \text{prox}_{\tilde{\lambda}h})(D(x^k - \gamma\nabla f(x^k)) + (I - \lambda DD^T)v^k - \lambda D y^k), & (3.4a) \\ y^{k+1} = (I - \text{prox}_{\tilde{\lambda}g})((x^k - \gamma\nabla f(x^k)) + (I - \lambda)y^k - \lambda D^T v^k), & (3.4b) \\ x^{k+1} = x^k - \gamma\nabla f(x^k) - \lambda D^T v^{k+1} - \lambda y^{k+1}. & (3.4c) \end{cases}$$

From the formula (3.3), we can easy obtain algorithm 1. So, the above algorithm is equivalent to apply directly PDFP<sup>2</sup>O of [1] to solve (3.2). According to the Theorem 3.1, we can obtain the convergence of Algorithm 1(SPDPF<sup>2</sup>O).  $\square$

Furthermore, we can analyze the convergence rate of Algorithm 1(SPDPF<sup>2</sup>O). Let  $u^k = (v^k, y^k, x^k)$  be a sequence obtained by algorithm SPDPF<sup>2</sup>O. Then the sequence  $u_k$  must converge to a point  $u^* = (v^*, y^*, x^*)$ , with  $x^*$  is a solution of problem (1.1), by the Theorem 3.7 of [1], we know the following estimate

$$\|x^k - x^*\| \leq \frac{c\theta^k}{1 - \theta},$$

where  $c = \|u^1 - u^0\|_\lambda$ ,  $\eta = \max\{\eta_1, \eta_2\}$ , with  $\eta_1$  and  $\eta_2$  given in condition 3.1 of [1].

From Lemma 2.2, the SPDPF<sup>2</sup>O iterates are generated by the action of a nonexpansive operator. Lemma 2.3 shows then that a stochastic coordinate descent version of the SPDPF<sup>2</sup>O converges towards a primal-dual point. This result will be exploited in two directions: first, we describe a stochastic minibatch algorithm, where a large dataset is randomly split into smaller chunks. Second, we develop an asynchronous version of the SPDPF<sup>2</sup>O in the context where it is distributed on a graph.



## 4 Application to stochastic approximation

### 4.1 Problem setting

Given an integer  $N > 1$ , consider the problem of minimizing a sum of composite functions

$$\inf_{x \in \mathcal{X}} \sum_{n=1}^N (f_n(x) + g_n(x)), \quad (4.1)$$

where we make the following assumption:

**Assumption 4.1.** *For each  $n = 1, \dots, N$ ,*

*(1)  $f_n$  is a convex differentiable function on  $\mathcal{X}$ , and its gradient  $\nabla f_n$  is  $1/\beta$ -Lipschitz continuous on  $\mathcal{X}$  for some  $\beta \in (0, +\infty)$ ;*

*(2)  $g_n \in \Gamma_0(\mathcal{X})$ ;*

*(3) The infimum of Problem (4.1) is attained;*

*(4)  $\cap_{n=1}^N \text{ri dom } g_n \neq \emptyset$ .*

This problem arises for instance in large-scale learning applications where the learning set is too large to be handled as a single block. Stochastic minibatch approaches consist in splitting the data set into  $N$  chunks and to process each chunk in some order, one at a time. The quantity  $f_n(x) + g_n(x)$  measures the inadequacy between the model (represented by parameter  $x$ ) and the  $n$ -th chunk of data. Typically,  $f_n$  stands for a data fitting term whereas  $g_n$  is a regularization term which penalizes the occurrence of erratic solutions. As an example, the case where  $f_n$  is quadratic and  $g_n$  is the  $l_1$ -norm reduces to the popular LASSO problem [5]. In particular, it is also useful to recover sparse signal.

### 4.2 Instantiating the SPDFP<sup>2</sup>O

We regard our stochastic minibatch algorithm as an instance of the SPDFP<sup>2</sup>O coupled with a randomized coordinate descent. In order to end that, we rephrase problem (4.1) as

$$\inf_{x \in \mathcal{X}^N} \sum_{n=1}^N (f_n(x) + g_n(x)) + \iota_{\mathcal{C}}(x), \quad (4.2)$$

where the notation  $x_n$  represents the  $n$ -th component of any  $x \in \mathcal{X}^N$ ,  $\mathcal{C}$  is the space of vectors  $x \in \mathcal{X}^N$  such that  $x_1 = \dots = x_N$ . On the space  $\mathcal{X}^N$ , we set  $f(x) = \sum_n f_n(x_n)$ ,  $g(x) = \sum_n g_n(x_n)$ ,  $h(x) = \iota_{\mathcal{C}}$  and  $D = I_{\mathcal{X}^N}$  the identity matrix. problem (4.2) is equivalent to

$$\min_{x \in \mathcal{X}^N} f(x) + g(x) + (h \circ D)(x). \quad (4.3)$$

We define the natural scalar product on  $\mathcal{X}^N$  as  $\langle x, y \rangle = \sum_{n=1}^N \langle x_n, y_n \rangle$ . Applying the SPDFP<sup>2</sup>O to solve problem (4.3) leads to the following iterative scheme:

$$\begin{aligned} z^{k+1} &= \text{proj}_{\mathcal{C}}(x^k - \gamma \nabla f(x^k) + (1 - \lambda)v^k - \lambda y^k), \\ v_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)v_n^k - \lambda y_n^k - z_n^{k+1}, \\ y_n^{k+1} &= (I - \text{prox}_{\tilde{\chi} g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda v_n^k), \\ x_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) - \lambda v_n^{k+1} - \lambda y_n^{k+1}, \end{aligned}$$

where  $\text{proj}_{\mathcal{C}}$  is the orthogonal projection onto  $\mathcal{C}$ . Observe that for any  $x \in \mathcal{X}^N$ ,  $\text{proj}_{\mathcal{C}}(x)$  is equivalent to  $(\bar{x}, \dots, \bar{x})$  where  $\bar{x}$  is the average of vector  $x$ , that is  $\bar{x} = N^{-1} \sum_n x_n$ . Consequently, the components of  $z^{k+1}$  are equal and coincide with  $\bar{x}^k - \gamma \nabla \bar{f}(\bar{x}^k) + (1 - \lambda)\bar{v}^k - \lambda \bar{y}^k$  where  $\bar{f}$ ,  $\bar{x}^k$ ,  $\bar{v}^k$  and  $\bar{y}^k$  are the averages of  $f$ ,  $x^k$ ,  $v^k$  and  $y^k$  respectively. By inspecting the  $v^k$   $n$ -update equation above, we notice that the latter equality simplifies even further by noting that  $\bar{v}^{k+1} = 0$  or, equivalently,  $\bar{v}^k = 0$  for all  $k \geq 1$  if the algorithm is started with  $\bar{v}^0 = 0$ . Finally, for any  $n$  and  $k \geq 1$ , the above iterations reduce to

$$\begin{aligned} \bar{x}^k - \gamma \nabla \bar{f}(\bar{x}^k) - \lambda \bar{y}^k &= \frac{1}{N} \sum_{n=1}^N (x_n^k - \gamma \nabla \bar{f}_n(x_n^k) - \lambda y_n^k), \\ v_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)v_n^k - \lambda y_n^k - (\bar{x}^k - \gamma \nabla \bar{f}(\bar{x}^k) - \lambda \bar{y}^k), \\ y_n^{k+1} &= (I - \text{prox}_{\tilde{\chi} g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda v_n^k), \\ x_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) - \lambda v_n^{k+1} - \lambda y_n^{k+1}. \end{aligned}$$

These iterations can be written more compactly as

---

**Algorithm 2** Minibatch SPDFP<sup>2</sup>O.

---

Initialization: Choose  $x^0, y^0 \in \mathcal{X}$ ,  $v^0 \in \mathcal{Y}$ , s.t.  $\sum_n v_n^0 = 0$ ,  $0 < \lambda \leq 1/2$ ,  $0 < \gamma < 2\beta$ .  
Do

- $\bar{x}^k - \gamma \nabla \bar{f}(\bar{x}^k) - \lambda \bar{y}^k = \frac{1}{N} \sum_{n=1}^N (x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k),$
  - For batches  $n = 1, \dots, N$ , do
$$\begin{aligned} v_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)v_n^k - \lambda y_n^k - (\bar{x}^k - \gamma \nabla \bar{f}(\bar{x}^k) - \lambda \bar{y}^k), \\ y_n^{k+1} &= (I - \text{prox}_{\lambda g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda v_n^k), \\ x_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) - \lambda v_n^{k+1} - \lambda y_n^{k+1}. \end{aligned} \tag{4.4}$$
  - Increment  $k$ .
- 

The following result is a straightforward consequence of Theorem 3.2.

**Theorem 4.1.** Suppose  $0 < \gamma < 2\beta$  and  $0 < \lambda \leq 1/2$ , and let Assumption 4.1 hold true. Then for any initial point  $(v^0, y^0, x^0)$  such that  $\bar{v}^0 = 0$ , the sequence  $\{\bar{x}^k\}$  generated by Minibatch SPDFP<sup>2</sup>O converges to a solution of problem (4.3).

At each step  $k$ , the iterations given above involve the whole set of functions  $f_n, g_n (n = 1, \dots, N)$ . Our aim is now to propose an algorithm which involves a single couple of functions  $(f_n, g_n)$  per iteration.

### 4.3 A stochastic minibatch splitting primal-dual fixed point algorithm

We are now in position to state the main algorithm of this section. The proposed stochastic minibatch splitting primal-dual fixed point algorithm (SMSPDFP<sup>2</sup>O) is obtained upon applying the randomized coordinate descent on the minibatch SPDFP<sup>2</sup>O:

---

**Algorithm 3** SMSPDFP<sup>2</sup>O.

---

Initialization: Choose  $x^0, y^0 \in \mathcal{X}$ ,  $v^0 \in \mathcal{Y}$ ,  $0 < \lambda \leq 1/2$ ,  $0 < \gamma < 2\beta$ .

Do

- Define  $\bar{x}^k - \gamma \nabla \bar{f}(\bar{x}^k) - \lambda \bar{y}^k = \frac{1}{N} \sum_{n=1}^N (x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k)$ ,  
 $\bar{v}^k = \frac{1}{N} \sum_{n=1}^N v_n^k$ ,
  - Pick up the value of  $\zeta^{k+1}$ ,
  - For batch  $n = \zeta^{k+1}$ , set
 
$$v_n^{k+1} = x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)v_n^k - \lambda y_n^k - (\bar{x}^k - \gamma \nabla \bar{f}(\bar{x}^k) - (1 - \lambda)\bar{v}^k - \lambda \bar{y}^k), \quad (4.5a)$$

$$y_n^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda v_n^k), \quad (4.5b)$$

$$x_n^{k+1} = x_n^k - \gamma \nabla f_n(x_n^k) - \lambda v_n^{k+1} - \lambda y_n^{k+1}. \quad (4.5c)$$
  - For all batches  $n \neq \zeta^{k+1}$ ,  $v_n^{k+1} = v_n^k, y_n^{k+1} = y_n^k, x_n^{k+1} = x_n^k$ .
  - Increment  $k$ .
- 

**Assumption 4.2.** The random sequence  $(\zeta^k)_{k \in \mathbb{N}^*}$  is i.i.d. and satisfies  $\mathbb{P}[\zeta^1 = n] > 0$  for all  $n = 1, \dots, N$ .

**Theorem 4.2.** Suppose  $0 < \gamma < 2\beta$  and  $0 < \lambda \leq 1/2$ , and let Assumption 4.1 and 4.2 hold true. Then for any initial point  $(v^0, y^0, x^0)$ , the sequence  $\{\bar{x}^k\}$  generated by SMSPDFP<sup>2</sup>O converges to a solution of problem (4.3).

*Proof.* Let us define the functions  $f$ ,  $g$ , and  $h$  are the ones defined in Section 4.2 and  $D = I_{\mathcal{X}^N}$ . Then the iterates  $((v_n^{k+1})_{n=1}^N, (y_n^{k+1})_{n=1}^N, (x_n^{k+1})_{n=1}^N)$  described by Equations (4.4) coincide with the iterates  $(v^{k+1}, y^{k+1}, x^{k+1})$  described by Equations (3.4). If we write these equations more compactly as  $(\tilde{v}^{k+1}, x^{k+1}) = T(\tilde{v}^k, x^k)$  where  $\tilde{v}^k = (v^k, y^k)^T$ , the operator  $T$  acts in the space  $\mathcal{V} = \mathcal{X}^N \times \mathcal{X}^N \times \mathcal{X}^N$ , then Lemma 2.2 shows that  $T$  is nonexpansive. Defining the selection operator  $\mathcal{S}_n$  on  $\mathcal{V}$  as  $\mathcal{S}_n(\tilde{v}, x) = (\tilde{v}_n, x_n)$ , we obtain that  $\mathcal{V} = \mathcal{S}_1(\mathcal{V}) \times \dots \times \mathcal{S}_N(\mathcal{V})$  up to an element reordering. To be compatible with the notations of Definition 2.4, we assume that  $J = N$  and that the random sequence  $\zeta^k$  driving the SMSPDFP<sup>2</sup>O algorithm is set valued in  $\{\{1\}, \dots, \{N\}\} \subset 2^{\mathcal{J}}$ . In order to establish Theorem 4.2, we need to show that the iterates  $(\tilde{v}^{k+1}, x^{k+1})$  provided by the SMSPDFP<sup>2</sup>O algorithm are those who satisfy the equation  $(\tilde{v}^{k+1}, x^{k+1}) =$

$T^{(\zeta^{k+1})}(\tilde{v}^k, x^k)$ . By the direct application of Lemma 2.3, we can obtain Theorem 4.2. If we write  $(\tilde{\delta}^{k+1}, \sigma^{k+1}) = T(\tilde{v}^k, x^k)$  where  $\tilde{\delta}^{k+1} = (\mu^{k+1}, \nu^{k+1})^T$ , then by Eq. (3.4a),

$$\mu_n^{k+1} = x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)v_n^k - \lambda y_n^k - (\bar{x}^k - \gamma \nabla f(\bar{x}^k) - (1 - \lambda)\bar{v}^k - \lambda \bar{y}^k) n = 1, \dots, N.$$

Observe that in general,  $\bar{v}^k \neq 0$  because in the SMSPDFP<sup>2</sup>O algorithm, only one component is updated at a time. If  $\{n\} = \zeta^{k+1}$ , then  $v_n^{k+1} = \mu_n^{k+1}$  which is Eq. (4.5a). All other components of  $v^k$  are carried over to  $v^{k+1}$ .

By Equation (3.4b) and (3.4c) we also get

$$\begin{aligned} \nu_n^{k+1} &= (I - \text{prox}_{\tilde{\chi}g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda v_n^k), \\ \sigma_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) - \lambda v_n^{k+1} - \lambda y_n^{k+1}. \end{aligned}$$

If  $\{n\} = \zeta^{k+1}$ , then  $y_n^{k+1} = \nu_n^{k+1}$ ,  $x_n^{k+1} = \sigma_n^{k+1}$  can easily be shown to be given by (4.5b) and (4.5c).

□

## 5 Distributed optimization

Consider a set of  $N > 1$  computing agents that cooperate to solve the minimization problem (4.1). Here,  $f_n, g_n$  are two private functions available at agent  $n$ . Our purpose is to introduce a random distributed algorithm to solve (4.1). The algorithm is asynchronous in the sense that some components of the network are allowed to wake up at random and perform local updates, while the rest of the network stands still. No coordinator or global clock is needed. The frequency of activation of the various network components is likely to vary.

The examples of this problem appear in learning applications where massive training data sets are distributed over a network and processed by distinct machines [5], [6], in resource allocation problems for communication networks [7], or in statistical estimation problems by sensor networks [8], [9].

## 5.1 Network model and problem formulation

We consider the network as a graph  $G = (Q, E)$  where  $Q = \{1, \dots, N\}$  is the set of agents/nodes and  $E \subset \{1, \dots, N\}^2$  is the set of undirected edges. We write  $n \sim m$  whenever  $n, m \in E$ . Practically,  $n \sim m$  means that agents  $n$  and  $m$  can communicate with each other.

**Assumption 5.1.**  *$G$  is connected and has no self loop.*

Now we introduce some notations. For any  $x \in \mathcal{X}^{|Q|}$ , we denote by  $x_n$  the components of  $x$ , i.e.,  $x = (x_n)_{n \in Q}$ . We regard the functions  $f$  and  $g$  on  $\mathcal{X}^{|Q|} \rightarrow (-\infty, +\infty]$  as  $f(x) = \sum_{n \in Q} f_n(x_n)$  and  $g(x) = \sum_{n \in Q} g_n(x_n)$ . So the problem (4.1) is equal to the minimization of  $f(x) + g(x)$  under the constraint that all components of  $x$  are equal.

Next we write the latter constraint in a way that involves the graph  $G$ . We replace the global consensus constraint by a modified version of the function  $\iota_C$ . The purpose of us is to ensure global consensus through local consensus over every edge of the graph.

For any  $\varepsilon \in E$ , say  $\varepsilon = \{n, m\} \in Q$ , we define the linear operator  $D_\varepsilon(x) : \mathcal{X}^{|Q|} \rightarrow \mathcal{X}^2$  as  $D_\varepsilon(x) = (x_n, x_m)$  where we assume some ordering on the nodes to avoid any ambiguity on the definition of  $D$ . We construct the linear operator  $D : \mathcal{X}^{|Q|} \rightarrow \mathcal{Y} \triangleq \mathcal{X}^{2|E|}$  as  $D(x) = (D_\varepsilon(x))_{\varepsilon \in E}$  where we also assume some ordering on the edges. Any vector  $y \in \mathcal{Y}$  will be written as  $y = (y_\varepsilon)_{\varepsilon \in E}$  where, writing  $\varepsilon = \{n, m\} \in E$ , the component  $y_\varepsilon$  will be represented by the couple  $y_\varepsilon = (y_\varepsilon(n), y_\varepsilon(m))$  with  $n < m$ . We also introduce the subspace of  $\mathcal{X}^2$  defined as  $\mathcal{C}_2 = \{(x, x) : x \in \mathcal{X}\}$ . Finally, we define  $h : \mathcal{Y} \rightarrow (-\infty, +\infty]$  as

$$h(y) = \sum_{\varepsilon \in E} \iota_{\mathcal{C}_2}(y_\varepsilon). \quad (5.1)$$

Then we consider the following problem:

$$\min_{x \in \mathcal{X}^{|Q|}} f(x) + g(x) + (h \circ D)(x). \quad (5.2)$$

**Lemma 5.1.** ([2]). *Let Assumptions 5.1 hold true. The minimizers of (5.2) are the tuples  $(x^*, \dots, x^*)$  where  $x^*$  is any minimizer of (4.1).*

## 5.2 Instantiating the SPDFP<sup>2</sup>O

Now we use the SPDFP<sup>2</sup>O to solve the problem (5.2). Since the newly defined function  $h$  is separable with respect to the  $(y_\varepsilon)_{\varepsilon \in E}$ , we get

$$\text{prox}_{\tau h}(y) = (\text{prox}_{\tau \iota_{C_2}}(y_\varepsilon))_{\varepsilon \in E} = ((\bar{y}_\varepsilon, \bar{y}_\varepsilon))_{\varepsilon \in E},$$

where  $\bar{y}_\varepsilon = (y_\varepsilon(n) + y_\varepsilon(m))/2$  if  $\varepsilon = \{n, m\}$ . With this at hand, the update equation (3.4a) of the SPDFP<sup>2</sup>O can be written as

$$z^{k+1} = ((\bar{z}_\varepsilon^{k+1}, \bar{z}_\varepsilon^{k+1}))_{\varepsilon \in E},$$

where

$$\begin{aligned} \bar{z}^{k+1} = & \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k + x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k}{2} \\ & + \frac{v_\varepsilon^k(n) + v_\varepsilon^k(m)}{2}, \end{aligned}$$

for any  $\varepsilon = \{n, m\} \in E$ . and  $d_n x_n$  coincides with the  $n$ -th component of the vector  $D^T D x$ ,  $d_n$  is the degree (i.e., the number of neighbors) of node  $n$ . Plugging this equality into Eq. (3.4a), it can be seen that  $v_\varepsilon^k(n) = -v_\varepsilon^k(m)$ . Therefore

$$\bar{z}^{k+1} = \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k + x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k}{2},$$

for any  $k \geq 1$ . Moreover

$$v_\varepsilon^{k+1} = \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k - (x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k)}{2} + v_\varepsilon^k(n).$$

From (3.4b) and (3.4c), the  $n^{\text{th}}$  component of  $y^{k+1}$  and  $x^{k+1}$  can be written

$$y_n^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda D^T v_n^k),$$

$$x_n^{k+1} = x_n^k - \gamma \nabla f_n(x_n^k) - \lambda D^T v_n^{k+1} - \lambda y_n^{k+1},$$

where for any  $v \in \mathcal{Y}$ ,

$$(D^T v)_n = \sum_{m: \{n, m\} \in E} v_{\{n, m\}}(n)$$

is the  $n$ -th component of  $D^T v \in \mathcal{X}^{|Q|}$ . Plugging Eq. (3.4b) and (3.4c) together with the expressions of  $\bar{z}_{\{n,m\}}^{k+1}$  and  $v_{\{n,m\}}^{k+1}$ , we can have

$$y_n^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} g_n})(x_n^k - \gamma \nabla f_n(x_n^k)) + (1 - \lambda)y_n^k - \lambda \sum_{m: \{n,m\} \in E} v_{\{n,m\}}^k(n),$$

$$\begin{aligned} x_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) - \lambda \sum_{m: \{n,m\} \in E} v_{\{n,m\}}^k(n) - \lambda y_n^{k+1} \\ &\quad - \lambda \sum_{m: \{n,m\} \in E} \left( \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k - (x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k)}{2} \right). \end{aligned}$$

The algorithm is finally described by the following procedure: Prior to the clock tick  $k+1$ , the node  $n$  has in its memory the variables  $x_n^k$ ,  $y_n^k$ ,  $\{v_{\{n,m\}}^k(n)\}_{m \sim n}$ ,  $\{x_m^k\}_{m \sim n}$  and  $\{y_m^k\}_{m \sim n}$ .

---

**Algorithm 4** Distributed SPDFP<sup>2</sup>O.

---

Initialization: Choose  $x^0, y^0 \in \mathcal{X}$ ,  $v^0 \in \mathcal{Y}$ , s.t.  $\sum_n v_n^0 = 0$ ,  $0 < \lambda \leq 1/(\lambda_{\max}(DD^T) + 1)$ ,  $0 < \gamma < 2\beta$ .

Do

- For any  $n \in Q$ , Agent  $n$  performs the following operations :

$$v_{\{n,m\}}^{k+1}(n) = \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k - (x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k)}{2} + v_{\{n,m\}}^k(n),$$

for all  $m \sim n$ ,

(5.3a)

$$y_n^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} g_n})(x_n^k - \gamma \nabla f_n(x_n^k)) + (1 - \lambda)y_n^k - \lambda \sum_{m: \{n,m\} \in E} v_{\{n,m\}}^k(n),$$
(5.3b)

$$\begin{aligned} x_n^{k+1} &= x_n^k - \gamma \nabla f_n(x_n^k) - \lambda \sum_{m: \{n,m\} \in E} v_{\{n,m\}}^k(n) - \lambda y_n^{k+1} \\ &\quad - \lambda \sum_{m: \{n,m\} \in E} \left( \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k - (x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k)}{2} \right). \end{aligned}$$
(5.3c)

- Agent  $n$  sends the parameter  $y_n^{k+1}, x_n^{k+1}$  to their neighbors respectively.
  - Increment  $k$ .
- 

**Theorem 5.1.** Suppose  $0 < \gamma < 2\beta$  and  $0 < \lambda \leq 1/(\lambda_{\max}(DD^T) + 1)$ , and let Assumption 4.1 and 5.1 hold true. Let  $u^k = (v^k, y^k, x^k)$  be the sequence generated by Distributed



*SPDFP<sup>2</sup>O* for any initial point  $(v^0, y^0, x^0)$ . Then for all  $n \in Q$  the sequence  $(x_n^k)_{k \in \mathbb{N}}$  converges to a solution of problem (4.1).

### 5.3 A Distributed asynchronous splitting primal-dual fixed point algorithm

In this section, we use the randomized coordinate descent on the above algorithm, we call this algorithm as distributed asynchronous splitting primal-dual fixed point algorithm (DSSPDFP<sup>2</sup>O). This algorithm has the following attractive property: at each iteration, a single agent, or possibly a subset of agents chosen at random, are activated. Moreover, if we let  $(\zeta^k)_{k \in \mathbb{N}}$  be a sequence of i.i.d. random variables valued in  $2^Q$ . The value taken by  $\zeta^k$  represents the agents that will be activated and perform a prox on their  $x$  variable at moment  $k$ . The asynchronous algorithm goes as follows:

---

#### Algorithm 5 DASPDFP<sup>2</sup>O.

---

Initialization: Choose  $x^0, y^0 \in \mathcal{X}$ ,  $v^0 \in \mathcal{Y}$ ,  $0 < \lambda \leq 1/(\lambda_{\max}(DD^T) + 1)$ ,  $0 < \gamma < 2\beta$ .

Do

- Select a random set of agents  $\zeta^{k+1} = \mathcal{B}$ .
  - For any  $n \in \mathcal{B}$ , Agent  $n$  performs the following operations :
    - For all  $m \sim n$ , do
 
$$v_{\{n,m\}}^{k+1}(n) = \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k - (x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k)}{2} + \frac{v_{\{n,m\}}^k(n) - v_{\{n,m\}}^k(m)}{2},$$
    - $y_n^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda \sum_{m: \{n,m\} \in E} v_{\{n,m\}}^k(n)),$
    - $x_n^{k+1} = x_n^k - \gamma \nabla f_n(x_n^k) + \lambda \sum_{m: \{n \sim m\} \in E} v_{\{n,m\}}^k(m) - \lambda y_n^{k+1} - \lambda \sum_{m: \{n,m\} \in E} (\frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k - (x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k)}{2}).$
    - For all  $m \sim n$ , send  $\{y_n^{k+1}, x_n^{k+1}, v_{\{n,m\}}^{k+1}(n)\}$  to Neighbor  $m$ .
  - For any agent  $n \notin \mathcal{B}$ ,  $y_n^{k+1} = y_n^k$ ,  $x_n^{k+1} = x_n^k$ , and  $v_{\{n,m\}}^{k+1}(n) = v_{\{n,m\}}^k(n)$  for all  $m \sim n$ .
  - Increment  $k$ .
-

**Assumption 5.2.** The collections of sets  $\{\mathcal{B}_1, \mathcal{B}_2, \dots\}$  such that  $\mathbb{P}[\zeta^1 = \mathcal{B}_i]$  is positive satisfies  $\bigcup \mathcal{B}_i = Q$ .

**Theorem 5.2.** Suppose  $0 < \gamma < 2\beta$  and  $0 < \lambda \leq 1/(\lambda_{\max}(DD^T) + 1)$ , and let Assumption 4.1, 5.1 and 5.2 hold true. Let  $(u_n^k)_{n \in Q} = (v_n^k, y_n^k, x_n^k)_{n \in Q}$  be the sequence generated by DASPDFP<sup>2</sup>O for any initial point  $(v^0, y^0, x^0)$ . Then the sequence  $x_1^k, \dots, x_{|Q|}^k$  converges to a solution of problem (4.1).

*Proof.* Let us define  $f, g, h$  and  $D$  are the ones defined in the problem 5.2. By Equations (3.4). We write these equations more compactly as  $(\tilde{v}^{k+1}, x^{k+1}) = T(\tilde{v}^k, x^k)$  where  $\tilde{v}^k = (v^k, y^k)^T$ , the operator  $T$  acts in the space  $\mathcal{V} = \mathcal{R} \times \mathcal{X}^{|Q|} \times \mathcal{X}^{|Q|}$ , and  $\mathcal{R}$  is the image of  $\mathcal{X}^{|Q|}$  by  $D$ . then by Lemma 2.2 we know  $T$  is nonexpansive. Defining the selection operator  $\mathcal{S}_n$  on  $\mathcal{V}$  as  $\mathcal{S}_n(\tilde{v}, x) = (\tilde{v}_\varepsilon(n)_{\varepsilon \in Q: n \in \varepsilon}, x_n)$ , where  $\tilde{v}_\varepsilon(n)_{\varepsilon \in Q: n \in \varepsilon} = (v_\varepsilon(n)_{\varepsilon \in Q: n \in \varepsilon}, y_n)^T$ . So, we obtain that  $\mathcal{V} = \mathcal{S}_1(\mathcal{V}) \times \dots \times \mathcal{S}_{|Q|}(\mathcal{V})$  up to an element reordering. Identifying the set  $\mathcal{J}$  introduced in the notations of Definition 2.4 with  $Q$ , the operator  $T^{(\zeta^k)}$  is defined as follows:

$$\mathcal{S}_n(T^{(\zeta^k)}(\tilde{v}, x)) = \begin{cases} \mathcal{S}_n(T(\tilde{v}, x)), & \text{if } n \in \zeta^k, \\ \mathcal{S}_n(\tilde{v}, x), & \text{if } n \neq \zeta^k. \end{cases}$$

Then by Lemma 2.3, we know the sequence  $(\tilde{v}^{k+1}, x^{k+1}) = T^{(\zeta^{k+1})}(\tilde{v}^k, x^k)$  converges almost surely to a solution of problem (1). Moreover, from Lemma 5.1, we have the sequence  $x^k$  converges almost surely to a solution of problem (4.1).

Therefore we need to show that the operator  $T^{(\zeta^{k+1})}$  is translated into the DASPDFP<sup>2</sup>O algorithm. If we write  $(\tilde{\delta}^{k+1}, \sigma^{k+1}) = T(\tilde{v}^k, x^k)$  where  $\tilde{\delta}^{k+1} = (\mu^{k+1}, \nu^{k+1})^T$ , then by Eq. (3.4a),

$$\mu_\varepsilon^{k+1} = \frac{x_n^k - \gamma \nabla f_n(x_n^k) - \lambda y_n^k - \lambda d_n v_n^k - (x_m^k - \gamma \nabla f_m(x_m^k) - \lambda y_m^k - \lambda d_m v_m^k)}{2} + \frac{v_\varepsilon^k(n) + v_\varepsilon^k(m)}{2}.$$

Getting back to  $(\tilde{v}^{k+1}, x^{k+1}) = T^{(\zeta^{k+1})}(\tilde{v}^k, x^k)$ , we have for all  $n \in \zeta^{k+1}$  and all  $m \sim n$ , then  $v_{\{n, m\}}^{k+1}(n) = \mu_{\{n, m\}}^{k+1}(n)$ . By Equation (3.4b) and (3.4c) we also get

$$\nu_n^{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} g_n})(x_n^k - \gamma \nabla f_n(x_n^k) + (1 - \lambda)y_n^k - \lambda D^T v_n^k),$$

$$\sigma_n^{k+1} = x_n^k - \gamma \nabla f_n(x_n^k) - \lambda D^T v_n^{k+1} - \lambda y_n^{k+1}.$$

Therefore, for all  $n \in \zeta^{k+1}$ , then  $y_n^{k+1} = \nu_n^{k+1}$ ,  $x_n^{k+1} = \sigma_n^{k+1}$ . If we use the identity  $(D^T v)_n = \sum_{m: \{n, m\} \in E} v_{\{n, m\}}(n)$  on the above equations, it can easy check these equations coincides with the  $x$ -update  $y$ -update in the DASPDFP<sup>2</sup>O algorithm.  $\square$

## 6 Numerical experiments

We consider the problem of  $l_1$ -regularized logistic regression. Denoting by  $m$  the number of observations and by  $q$  the number of features, the optimization problem writes

$$\inf_{x \in \mathbb{R}^q} \frac{1}{m} \sum_{i=1}^m \log(1 + e^{-y_i a_i^T x}) + \tau \|x\|_1, \quad (6.1)$$

where the  $(y_i)_{i=1}^m$  are in  $\{-1, +1\}$ , the  $(a_i)_{i=1}^m$  are in  $\mathbb{R}^q$ , and  $\tau > 0$  is a scalar. Let  $(\mathcal{W})_{n=1}^N$  indicate a partition of  $\{1, \dots, m\}$ . The optimization problem then writes

$$\inf_{x \in \mathbb{R}^q} \sum_{n=1}^N \sum_{i \in \mathcal{W}_n} \frac{1}{m} \log(1 + e^{-y_i a_i^T x}) + \tau \|x\|_1, \quad (6.2)$$

or, splitting the problem between the batches

$$\inf_{x \in \mathbb{R}^{Nq}} \sum_{n=1}^N \left( \sum_{i \in \mathcal{W}_n} \frac{1}{m} \log(1 + e^{-y_i a_i^T x_n}) + \frac{\tau}{N} \|x_n\|_1 \right) + \iota_{\mathcal{C}}(x), \quad (6.3)$$

where  $x = (x_1, \dots, x_N)$  is in  $\mathbb{R}^{Nq}$ . It is easy to see that problems (6.1), (6.2) and (6.3) are equivalent and problem (6.3) is in the form of (4.2).

### Acknowledgements

This work was supported by the National Natural Science Foundation of China (11131006, 41390450, 91330204, 11401293), the National Basic Research Program of China (2013CB 329404), the Natural Science Foundations of Jiangxi Province (CA201107114, 20114BAB 201004).

## References

- [1] Chen P J, Huang J G and Zhang X Q 2013 A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration *Inverse Problems* 29 025011-33.
- [2] Bianchi P, Hachem W and Iutzeler F 2014 A Stochastic coordinate descent primal-dual algorithm and applications to large-scale composite (arXiv:1407.0898v1 [math.OC] 3 Jul 2014) *Optimization*
- [3] Combettes P L and Wajs V R 2005 Signal recovery by proximal forward-backward splitting *Multiscale Model. Simul.* 4 1168-200.
- [4] Rudin L I, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms *Physica D* 60 259-68.
- [5] Forero, P A, Cano A and Giannakis G B 2010 Consensus-based distributed support vector machines *The Journal of Machine Learning Research* 99 1663-1707.
- [6] Agarwal A, Chapelle O, Dudík M, and Langford J 2011 A reliable effective terascale linear learning system arXiv preprint arXiv:1110.4198.
- [7] P. Bianchi and J. Jakubowicz, Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization, *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391- 405, February 2013.
- [8] S.S. Ram, V.V. Veeravalli, and A. Nedic, Distributed and recursive parameter estimation in parametrized linear state-space models, *IEEE Trans. on Automatic Control*, vol. 55, no. 2, pp. 488-492, 2010.
- [9] P. Bianchi, G. Fort, and W. Hachem, Performance of a distributed stochastic approximation algorithm, *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7405-7418, November 2013.
- [10] Lions P L and Mercier B 1979 Splitting algorithms for the sum of two nonlinear operators *SIAM J. Numer. Anal.* 16 964C79

- [11] Pssty G B 1979 Ergodic convergence to a zero of the sum of monotone operators in Hilbert space J. Math. Anal. Appl. 72 383C90
- [12] Moreau J-J 1962 Fonctions convexes duales et points proximaux dans un espace hilbertien C. R. Acad. Sci.,
- [13] Paris I 255 2897C99 Combettes P L and Wajs V R 2005 Signal recovery by proximal forward-backward splitting Multiscale Model. Simul. 4 1168C200
- [14] Micchelli C A, Shen L and Xu Y 2011 Proximity algorithms for image models: denoising Inverse Problems 27 45009C38
- [15] Yu. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, SIAM Journal on Optimization, vol. 22, no. 2, pp. 341C362, 2012.
- [16] O. Fercoq and P. Richtarik, Accelerated, parallel and proximal coordinate descent, arXiv preprint arXiv:1312.5799, 2013.
- [17] M. Bacak, The proximal point algorithm in metric spaces, Israel Journal of Mathematics, vol. 194, no. 2, pp. 689C701, 2013.